Australian Government
Department of Defence
Defence Science and
Technology Organisation

Net Warrior D10 Technology Report: Airborne Early Warning and Control (AEW&C) and Data Link Nodes

Derek Dominish, Peter Temple, Christos Sioutis, Callum Baillie and Kate Foster

Air Operations Division
Defence Science and Technology Organisation

DSTO-TR-2567

ABSTRACT

This report discusses the Air Operations Division contribution to the Net Warrior demonstration event NW-D10. NW-D10 was held in September 2010 and demonstrated information interoperability of middleware technologies in dynamic environments with real mission systems. An overview of the NW-D10 event is provided along with a discussion of the technologies demonstrated. The outcomes of NW-D10 are presented and future Net Warrior events and enabling research are outlined.

RELEASE LIMITATION

Approved for public release

Published by

Air Operations Division
DSTO Defence Science and Technology Organisation
506 Lorimer St
Fishermans Bend, Victoria 3207 Australia

Telephone: (03) 9626 7000
Fax: (03) 9626 7999

AR-015-030
April 2012

APPROVED FOR PUBLIC RELEASE

Net Warrior D10 Technology Report: Airborne Early Warning and Control (AEW&C) and Data Link Nodes

Executive Summary

The Defence Science and Technology Organisation (DSTO) Net Warrior initiative is aligned with the Australian Defence Organisation's (ADO) approach to implementing Network Centric Warfare (NCW) through 'learning by doing'. Net Warrior was conceived to address, through experimentation, new and evolving net centric capabilities and mission system integration technologies to enhance Australian Defence Force's (ADF) joint war fighting capabilities. Net Warrior experimentation is conducted with real systems, testbeds and simulators across DSTO and enables the organisation to provide advice to the ADO regarding the extent to which it needs to consider and implement particular NCW concepts and technologies. This report discusses the Air Operations Division's (AOD) contribution to the Net Warrior demonstration event *NW-D10*, which supported aspects of tasks CIO 07/042 (Tactical Data Links) and DMO 07/044 (Wedgetail).

NW-D10 was held in September 2010 and demonstrated information interoperability of middleware technologies in dynamic environments with real mission systems. It attracted representatives from many parts of the ADO, including Capability Development Group (CDG), Chief Information Officer Group (CIOG), the Defence Materiel Organisation (DMO), Headquarters Joint Operations Command (HQJOC), Surveillance and Response Group (SRG), Air Force Headquarters (AFHQ) and DSTO management.

NW-D10 involved integrating laboratory-based systems that were high fidelity representations containing both operational and simulated elements. These systems were distributed across different buildings and developed for different environments, specifically air, tactical data links, command and control, and intelligence, surveillance and reconnaissance. The AOD contribution to NW-D10 was the Wedgetail Integration Research Environment (WIRE) and the Airborne Systems Connectivity Environment Laboratory (ASCEL). The WIRE is an AOD capability developed to support Wedgetail Airborne Early Warning and Control (AEW&C) acquisition, in-service support and capability enhancement. It comprises the mission computing operational flight program from Wedgetail, together with a stimulation environment and additional software components developed by AOD. The ASCEL provides a capability to investigate tactical data link integration through the use of commercial off-the-shelf (COTS) equipment for scenario generation, track forwarding, network management, and displaying and recording data.

NW-D10 fostered a small community of cross disciplinary applied research that is of operational significance. It demonstrated a way forward for enhanced situation pictures for Regional Operations Centres (ROC), Wedgetail and other tactical and command and control nodes and generated a multitude of ideas for future collaborative work. The NW-D10 event demonstrated the utility of applying modern middleware technologies to integrate mission and combat systems into a seamless networked informational environment. Systems connected within this networked environment can become aware of and contribute to information that may enhance individual decision support functions. It is when these systems adapt to new information sources that the warfighter, who is becoming increasingly dependant on these systems, can gain a fundamental information advantage over an adversary.

The NW-D10 event involved the integration of a number of technologies and methodologies, including component based systems; service oriented architectures (SOA); middleware architectures, adapters and gateways; frameworks; tactical data links; and visualisation tools. Seamless information interoperability between disparate systems was demonstrated. Achieving this information interoperability was a non-trivial research task that was led by AOD. It required the definition of a common data ontology for information translation and the development of adaptors and gateways used to integrate the flow of information between the systems. The NW-D10 outcomes for AOD were:

- deeper understanding of tactical NCW integration challenges through a ‘learning by doing’ approach;
- enabling research that developed the understanding of the application of SOA methodologies to the domain to manage complexity, risk and to bridge domains;
- facilitation of systems development and integration via the reuse of system services;
- demonstration to ADO stakeholders of adaptation and integration technologies for seamless information interoperability between disparate tactical and enterprise systems; and
- successful collaboration across multiple divisions to develop and demonstrate NCW concepts using the Net Warrior battlelab network infrastructure.

The AOD Net Warrior community is continuing to investigate the utility of applying modern distributed object computing technologies to the networked systems integration domain. Events in the near future will focus on the integration of testbeds that are representative of airborne mission systems contained on the Wedgetail and Joint Strike Fighter platforms. In support of this, the AOD Net Warrior community has an enabling research program where emerging technologies and capabilities can be evaluated within the context of an informational network of real systems. The emerging technologies that are currently being investigated by AOD are: agent technologies applied to distributed object computing and SOA environments for automated decision support; mobile technologies with a tactical sensor suite and situational communication endpoint; CORBA Component Model architectures for avionics mission systems integration; and distributed mission training to investigate the next generation of air force training systems.

Authors

Derek Dominish
Air Operations Division

Mr Derek Dominish is a senior airborne integration analyst in the Airborne Mission Systems branch within the Air Operations Division at DSTO. He has extensive experience over 20 years in operating systems, architecting large-scale commercial and Defence programs and middleware-based systems. Derek currently leads the Net Warrior initiative at DSTO, which is tasked with research into middleware-based SOA systems integration.

Peter Temple
Air Operations Division

Mr Peter Temple is a senior research engineer in the Airborne Mission Systems branch within the Air Operations Division at DSTO. He has a Bachelor of Engineering (Electrical) (Hons) and a Bachelor of Science (Mathematics) from Adelaide University, and has worked for 22 years in the field of airborne mission system integration. Peter's current work includes acquisition and through-life support for the Wedgetail Airborne Early Warning and Control capability, and research related to mission system integration and performance.

Christos Sioutis
Air Operations Division

Dr Christos Sioutis is a research scientist in the the Airborne Mission Systems branch within the Air Operations Division at DSTO. Christos's research interests are currently focused in the areas of software architecture, artificial intelligence and tactical data links.

Callum Ballie
Air Operations Division

Mr Callum Baillie graduated with a Bachelor of Engineering (Software Engineering) (Hons) from the University of Adelaide in 2008. In 2009 he worked for CAE Professional Services and was contracted to the Airborne Mission Systems branch of the Air Operations Division at DSTO to support research into the Joint Strike Fighter. In 2010, Callum joined the Airborne Mission Systems branch and for the last two years has worked on tactical data links and acquisition and through-life support for the Wedgetail Airborne Early Warning and Control capability.

Kate Foster
Air Operations Division

Dr Kate Foster is a mission system software scientist within the Airborne Mission Systems branch of the Air Operations Division at DSTO. Her current work involves supporting the Wedgetail Airborne Early Warning and Control capability, participating in the Net Warrior initiative to investigate aspects of net centricity, and research into smartphone integration in tactical environments and middleware-based SOA system integration. Kate has a Bachelor of Engineering (Electrical and Electronics) (Hons) and a PhD from Swinburne University.

Contents

Index of Figures

Glossary

ADF	Australian Defence Force
ADGESIM	Air Defence Ground Environment SIMulator
ADIIB	Australian ISR Integration Backbone
ADO	Australian Defence Organisation
AEW&C	Airborne Early Warning and Control
AFHQ	Air Force Headquarters
AI	Artificial Intelligence
AMI	Asynchronous Message Invocation
ANSI	American National Standards Institute
AOD	Air Operations Division
AOSC	Air Operations Simulation Centre
API	Application Programming Interface
ARMS	Amalgamated Remote Management System
ASCEL	Airborne Systems Connectivity Environment Laboratory
AWD	Air Warfare Destroyer
BLOS	Beyond Line of Sight
C3ID	Command Control Communications and Intelligence Division
CCM	CORBA Component Model
CDG	Capability Development Group
CGF	Computer Generated Forces
CIOG	Chief Information Officer Group
COP	Common Operating Picture
CORBA	Common Object Request Broker Architecture
COTS	Commercial-Off-The-Shelf
DACS	Desktop Aircraft Cockpit Simulator
DDS	Data Distribution Service
DII	Dynamic Invocation Interface
DMO	Defence Materiel Organisation
DOC	Distributed Object Computing

DoD	Department of Defense
DSI	Dynamic Skeleton Interface
DSTO	Defence Science and Technology Organisation
FCA	Frequency Clearance Agreement
FOCAL	Future Operations Centre Analysis Laboratory
GIG	Global Information Grid
GIOP	General Inter ORB Protocol
GM	Gateway Manager
HQJOC	Headquarters Joint Operations Command
IDL	Interface Description Language
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IIOP	Internet Inter-ORB Protocol
IR	Interface Repositories
ISO	International Organization for Standardization
ISR	Intelligence Surveillance and Reconnaissance
ISRAIL	ISR Analysis and Integration Laboratory
ISRD	Intelligence Surveillance and Reconnaissance Division
JDBC	Java Data-Base Connectivity
JDO	Java Data Objects
JREAP	Joint Range Extension Applications Protocol
LVC	Live, Virtual and Constructive
MIDS	Multifunctional Information Distribution System
MOD	Maritime Operations Division
MSCT	Multi Source Correlator Tracker
MST	Mission System Testbed
NCOIC	Network Centric Operations Industry Consortium
NCW	Network Centric Warfare
NCWIIS	NCW Integration and Implementation Strategy
NMS	Network Management System
OACE	Open Architecture Common Operating Environment

OMA	Object Management Architecture
OMG	Object Management Group
ORB	Object Request Broker
PPLI	Precise Participant Location and Identification
QoS	Quality-of-Service
RAAF	Royal Australian Air Force
RF	Radio Frequency
RIPS	Radar and Identify Friend Foe Performance Simulator
ROC	Regional Operations Centre
SOA	Service Oriented Architecture
SRG	Surveillance and Response Group
SRL	System Readiness Level
TBF	Time Based Filter
TDF	Tactical Display Framework
TDL	Tactical Data Link
TIA	Telecommunications Industry Association
TSDF	Time Slot Duty Factor
US	United States
WIRE	Wedgetail Integration Research Environment

This page is intentionally blank

1 Introduction

In alignment with the Australian Defence Organisation's (ADO) approach to implementing Network Centric Warfare (NCW) through 'learning by doing' [Department of Defence 2009], the Defence Science and Technology Organisation (DSTO) Net Warrior Initiative was conceived to address, through experimentation, new and evolving net centric capabilities and mission system integration technologies to enhance Australian Defence Force (ADF) joint war fighting capabilities.

Net Warrior experimentation is conducted with high fidelity systems, testbeds and simulators across DSTO and, in future, wider Defence. Such experimentation may be applied to the operational, systems and technical elements of NCW. It will enable DSTO to provide advice to the ADO regarding the extent to which it needs to consider and implement particular NCW concepts and technologies. The overall purpose of the Net Warrior exercises is to contribute to the mitigation of risk to acquisition and implementation of NCW through the conduct of multi-nodal and multi-purpose exercises. The aim of Net Warrior events is to showcase legacy and future platforms and technologies working in tandem in support of NCW operations ideals. The primary challenge faced by Net Warrior is the horizontal bridging of middleware technologies that have been designed in isolation from one another and implemented for operation in differing environments.

Net Warrior experimentation is conducted through a program of events categorised as infrastructure events (NW-I#), demonstration events (NW-D#) and experimentation events (NW-E#). Infrastructure events demonstrate serviceable and operating infrastructure. Demonstration events show how technologies well established in other domains could be employed by the ADO. The purpose of experimentation events is to advance the understanding of how the use of particular technologies and systems impacts operational effectiveness. This involves attempting to find a link between cause and effect and obtaining metrics to quantify operational effectiveness.

This report discusses the Air Operations Division (AOD) contribution to the Net Warrior demonstration event *NW-D10*, which supported aspects of tasks CIO 07/042 (Tactical Data Links) and DMO 07/044 (Wedgetail). Section 2 describes the NW-D10 demonstration and provides an overview of each node and their interaction. Section 3 discusses the technologies that underpinned the AOD nodes. Section 4 presents the outcomes from NW-10 and Section 5 describes the enabling research being conducted by AOD in support of future events.

2 Overview of the NW-D10 Event

NW-D10 was held in September 2010 and aimed to demonstrate information interoperability of middleware technologies in dynamic environments with high fidelity representations of mission systems. A potential airborne threat was detected and averted in a mock homeland security scenario. The NW-D10 demonstration was conducted three times in September 2010 and attracted representation from many parts of the ADO, including Capability Development Group (CDG), Chief Information Officer Group (CIOG), the Defence Materiel Organisation (DMO), Headquarters Joint Operations Command (HQJOC), Surveillance and Response Group (SRG), Air Force Headquarters (AFHQ) and DSTO management.

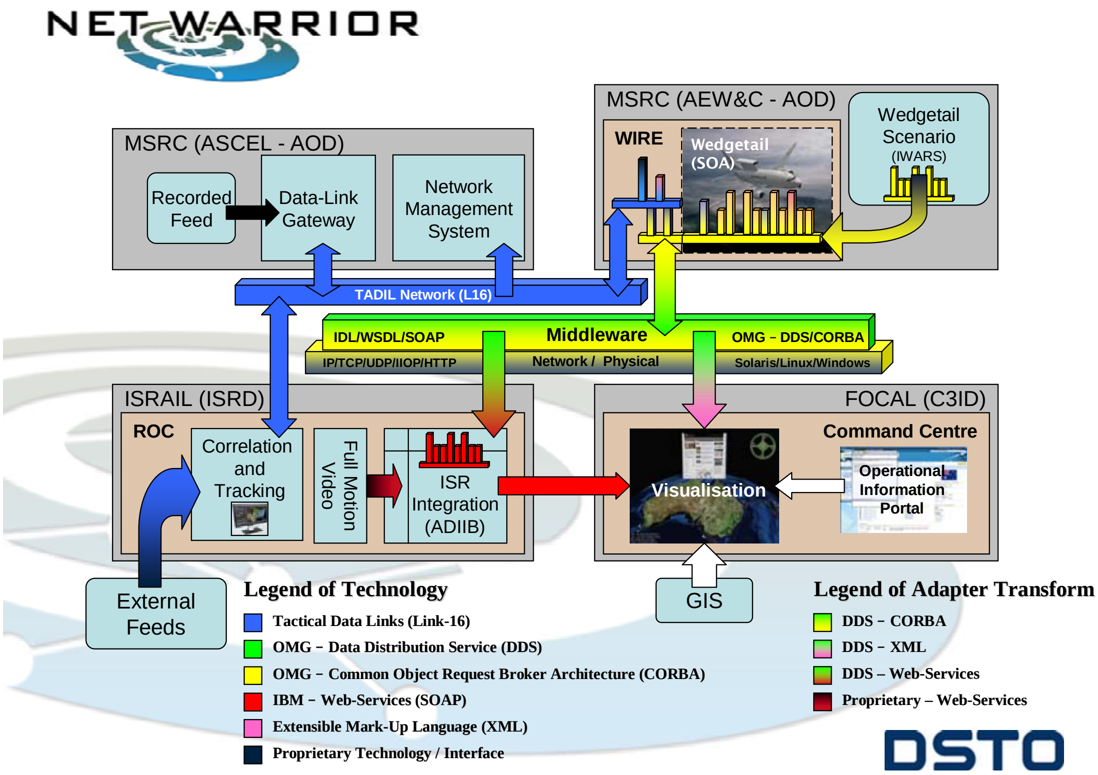


Figure 1: NW-D10 systems and information flow

Figure 1 illustrates the systems that contributed to NW-D10 and the interconnectedness within the network. NW-D10 involved integrating laboratory-based systems that were high fidelity representations containing both operational and simulated elements. These systems were distributed across different buildings and developed for different environments, specifically air, tactical data links, command and control, and intelligence, surveillance and reconnaissance (ISR). Partners in NW-D10 were:

- AOD, with the Wedgetail Integration Research Environment (WIRE) and the Airborne Systems Connectivity Environment Laboratory (ASCEL);
- Command Control Communications and Intelligence Division (C3ID), with the Future Operations Centre Analysis Laboratory (FOCAL); and
- Intelligence Surveillance and Reconnaissance Division (ISRD), with the ISR Analysis and Integration Laboratory (ISRAIL).

The AOD contribution to NW-D10 is described in the following two sections.

2.1 Wedgetail Integration and Research Environment

The WIRE is an AOD capability developed to support Wedgetail Airborne Early Warning and Control (AEW&C) acquisition [Defence Materiel Organisation 2009], in-service support and capability enhancement. It comprises the mission computing operational flight program from Wedgetail, together with a stimulation environment and additional software components[1] developed by DSTO. This software runs in a laboratory environment on commercial-off-the-shelf (COTS) workstations that are functionally equivalent to the Wedgetail mission computers. The WIRE is used for performance analysis, evaluation and familiarisation with the Wedgetail human-machine interface, and exploration of the integration of new technologies into the platform.

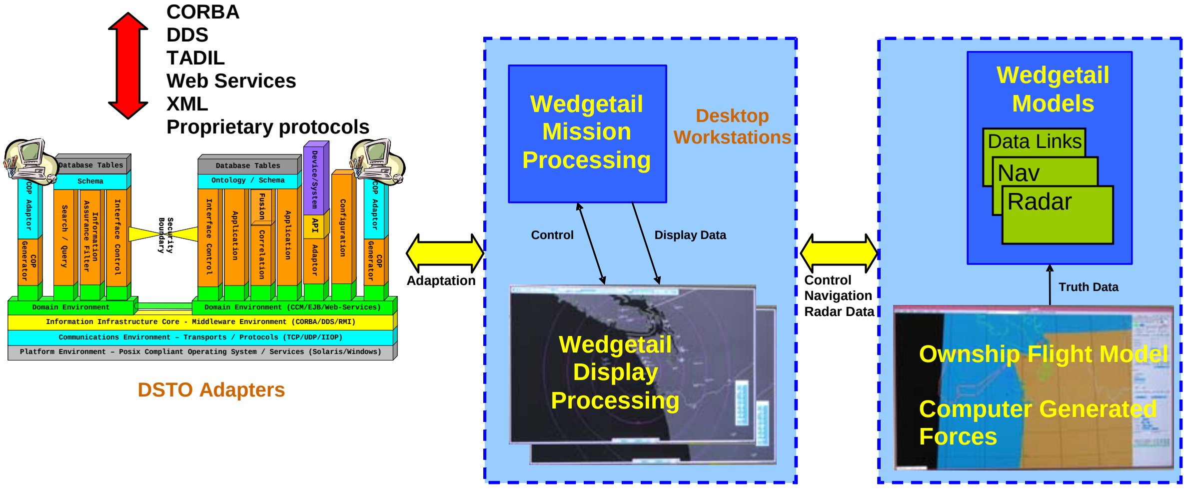


Figure 2: The Wedgetail Integration Research Environment for NW-D10

Figure 2 illustrates the WIRE architecture as used in NW-D10. Wedgetail Mission Processing provides all of the mission control, multi-sensor integration and tactical data link processing. Mission Processing feeds up to 10 displays that replicate the console displays in the aircraft. The Wedgetail software is stimulated in the same way as in the

[1] A description of these software components can be found in Section 3.2 and [Foster et al. 2007; Sioutis et al. 2008].

operational environment through partially validated models of the aircraft hardware. The most critical model is the Radar and Identify Friend Foe Performance Simulator (RIPS). RIPS models the radar detection performance and other radar functions such as control and calibration. The models in turn are stimulated with truth data that is generated by an ownship flight model and computer generated forces (CGF) acting on a scripted scenario. For NW-D10, the scenario did not dynamically interact with the scenarios scripted for the other participating nodes. The important addition to the WIRE for NW-D10 was the utilisation of middleware services to provide information interoperability with external systems (shown by the red arrow in Figure 2). This capability was provided by the Mission System Testbed (MST) [Foster et al. 2007], which is an AOD developed middleware based gateway for adaptation to external systems.

2.2 Airborne Systems Connectivity Environment Laboratory

Due to the sheer size and complexity of tactical data link (TDL) standards, there exists considerable risk that implementations from different vendors are not fully interoperable. One way to de-risk this integration is through live testing using the systems in question. The ASCEL [Filippidis, Doan & Tobin 2007; Filippidis & Gencarelli 2008; Zalcman et al. 2006] is used to provide this function (as well as other TDL research) through the use of COTS equipment for scenario generation, track forwarding, network management and displaying/recording of TDL data. This controlled laboratory environment provides the means to connect and translate across multiple arbitrary TDL types, stimulate connected systems using finely controlled TDL scenarios, monitor and capture the network traffic for analysis.

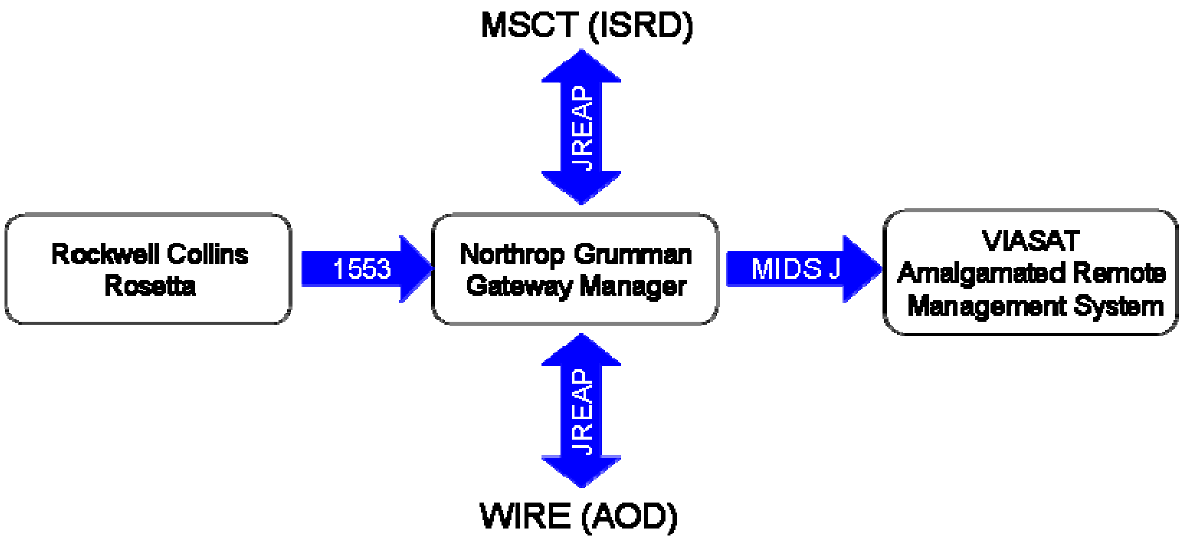


Figure 3: NW-D10 data link connectivity

Figure 3 illustrates the connectivity of TDL equipment in the ASCEL for NW-D10. The Northrop Grumman Gateway Manager (GM) was used to connect to different types of

TDL and seamlessly translate between them. The vertical axis is a simplified representation of how external, bi-directional, beyond line of sight (BLOS) Link-16 connectivity was achieved through the Joint Range Extension Applications Protocol (JREAP-C).

The horizontal axis represents connectivity of systems internal to the ASCEL. The Rockwell Collins Rosetta[2] system was used to generate a small number of scripted Link 16 tracks that were periodically transmitting Precise Participant Location and Identification (PPLI) messages, effectively simulating a small Link 16 network. Figure 4 illustrates Rosetta executing a scripted scenario. Rosetta was connected to the GM using a MIL-STD-1553 avionics real time bus.

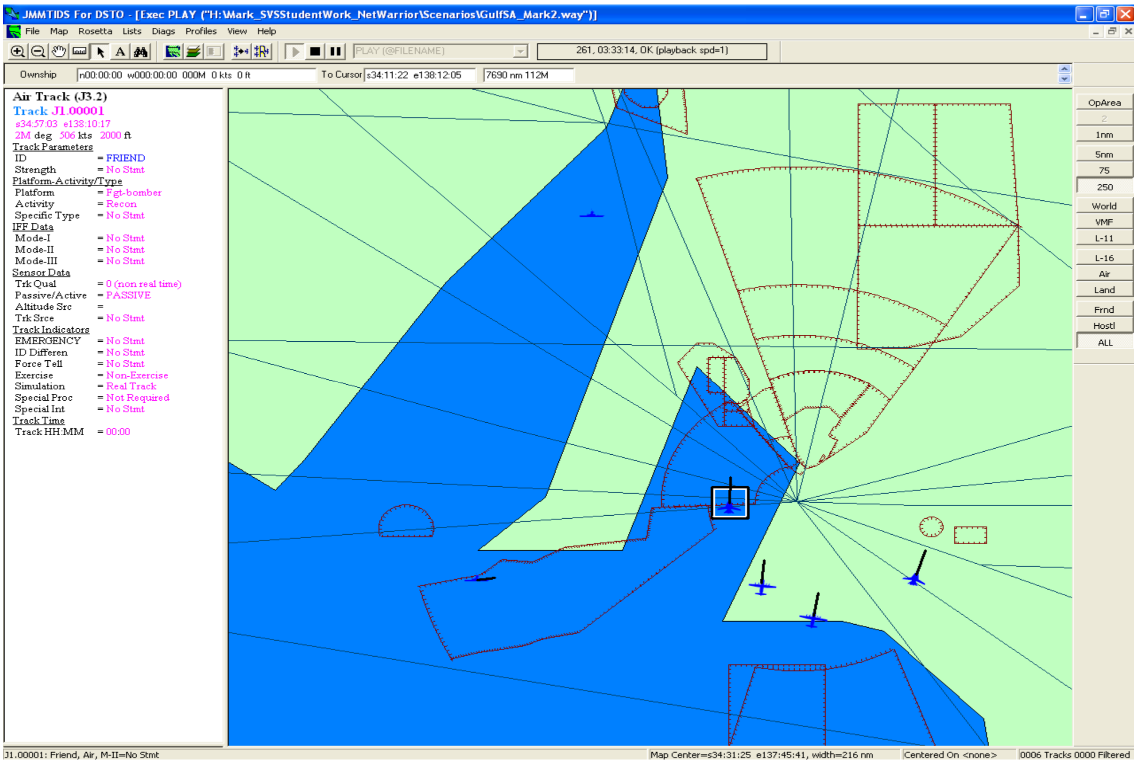


Figure 4: Rockwell Collins Rosetta system

The NW-D10 scenario combined a number of assets, some of which were real (provided by the Multi Source Correlator Tracker (MSCT)) and the others scripted (provided through scenarios in Rosetta and the WIRE). All assets were assigned specific Link 16 identification numbers and mapped to a Link 16 network design. This was loaded into the VIASAT Amalgamated Remote Management System (ARMS) and was used to monitor the Link 16 network characteristics. Of particular interest was the time slot duty factor (TSDF) utilisation of the network which must fall within the required aviation frequency clearance

---

[2] More detailed information about Rosetta is available from the Rosetta Technology website <http://www.rockwellcollins.com/sitecore/content/Data/Products/Communications_and_Networks/Networks/Rosetta_Technology.aspx>.

agreement (FCA) for use of Link 16 in Australia. Figure 5 illustrates the ARMS system being used to monitor the TSDF utilisation of the network. The two graphs allow comparison of the planned versus observed utilisation and all data is recorded for post analysis.

In a typical operational deployment the ARMS system connects to a Multifunctional Information Distribution System (MIDS) terminal to monitor the radio frequency (RF) traffic of a Link 16 network. The ASCEL does not have a MIDS terminal but the GM can be configured to emulate it. ARMS operated with the GM in the same way that it would have done so using a real MIDS terminal.

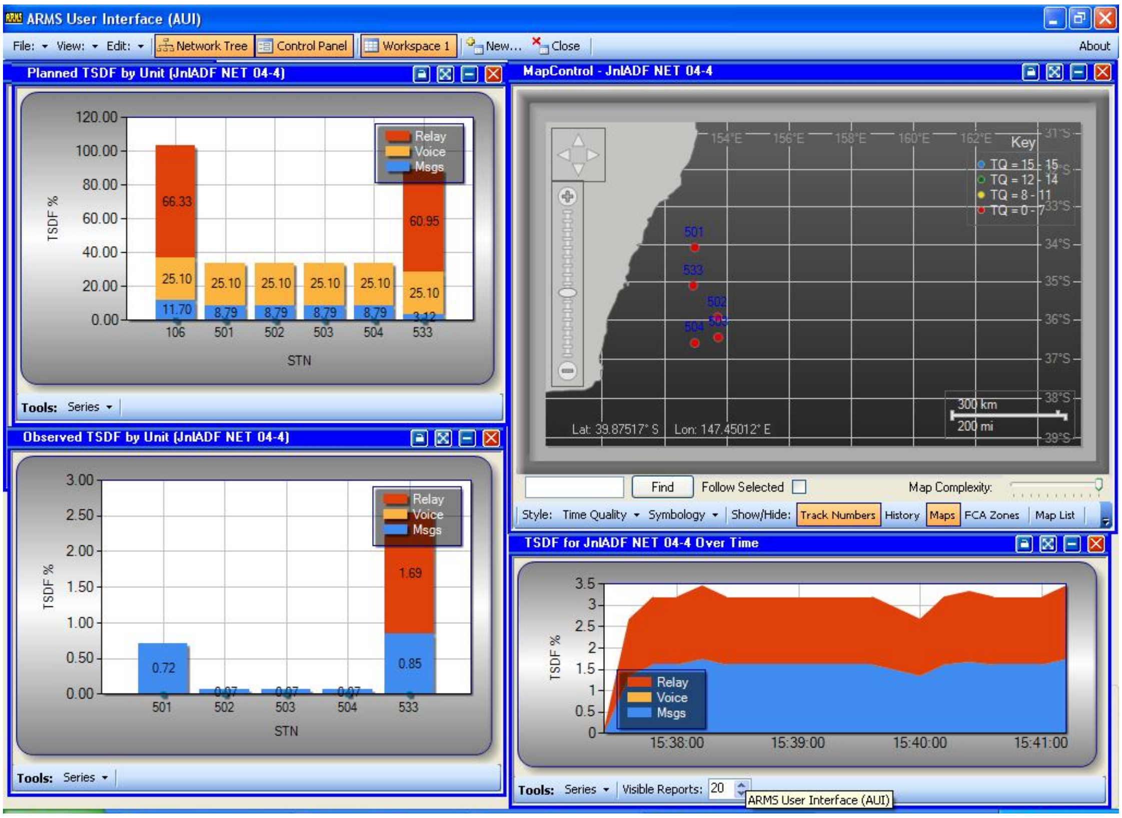


Figure 5: VIASAT Amalgamated Remote Management System

2.3 NW-D10 Information Flows and Situation Picture

NW-D10 had four information sources contributing to the scenario:

1) The WIRE contributed a detailed surveillance picture of a particular area of interest. Its stimulation environment provided a number of scripted targets which constituted the main NW-D10 scenario and additionally a number of background civil aviation tracks.

2) The ASCEL contributed a small number of scripted tracks transmitting messages which represented a Link 16 network in close proximity to the scenario.

3) The ISRAIL contributed a number of tracks forwarded from live feeds external to DSTO. These were live real world tracks which were merged with the simulated ones. The purpose of this was to demonstrate that the systems used in NW-D10 were high fidelity systems containing operational elements (not purely simulators), hence providing a real world fidelity to the demonstration.

4) The ISRAIL also contributed ISR information which was used to inform decisions during the scenario. Examples include detailed data on civil tracks (e.g. flight plans) and full motion video.

The primary consumer of all information was FOCAL which combined all sources into a common operating picture (COP). FOCAL hosted the NW-D10 event during the main demonstration and provided the environment of an operations centre designed to give commanders better situational awareness, mission planning and decision making capabilities.

A key technical aspect identified early in NW-D10 discussions was that each of the domain environments was to establish an architecture upon which many of their applications were to be executed. Each environment was specifically tuned for its respective domain and presented different interface and performance characteristics. The middleware implementations used in NW-D10 included the Common Object Request Broker Architecture (CORBA) [Object Management Group 2000, 2008] and Data Distribution Service (DDS) [Object Management Group 2007] originating from AOD; the Australian ISR Integration Backbone (ADIIB), based on Web Services technology, originating from ISRD; and real time XML originating from C3ID.

Achieving this information interoperability required the definition of common data ontology for information translation, development of adaptors, bridges and gateways to integrate the informational flow through to connected systems, and workflow integration. The connected systems operated as normal but became attuned to additional information available within the combined middleware network. Tracks were derived from multiple sources and sent to visualisation devices and systems across the network (including across Net Warrior nodes).

3 Demonstrated Technologies

Net centric environments are underpinned by a range of standards and technologies. Such technologies that were important to NW-D10 include component based systems, service oriented architecture (SOA), middleware and frameworks. Component based architectures, supported by middleware and built on top of frameworks are able to satisfy design needs of applications to produce stable mission and net centric systems. NW-D10 employed a SOA approach to integration, with common software component mechanisms, interfaces and adapters encapsulated within a patterned framework. The underlying middleware infrastructure environment provides and manages access to resources, such as

communications, services and the machinery that provides the capabilities for location independence and agility. SOA concepts, when applied to the needs of net centricity, are able to achieve flexible and adaptable operational effectiveness through the integration of disparate systems and capabilities.

3.1 Middleware Architecture

Through the Object Management Group (OMG) standardisation of the Distributed Object Computing (DOC) technologies of CORBA, DDS and CORBA Component Model (CCM) can be applied to mission critical military SOA systems. These systems often have a broad range of middleware infrastructure requirements, such as using middleware to port system components to multiple computing and communication environments. Furthermore, they allow applications and components to communicate effectively as peers within a distributed quality-of-service (QoS) rich environment.

The DDS publish and subscribe model is considered complementary to the DOC client/server model provided by CORBA. CORBA allows for multiple computers to be used to distribute processing. This capability is best suited to applications in which one or more software components (servants) collaborate to supply a service to one or more other interacting components (clients). Furthermore, these component interactions are often orchestrated within the principals and practice of a SOA environment. The DDS standard can be used to enhance these interactions by facilitating the sharing of data as *topics* across multiple computers, and possibly between distinctly separate applications. Therefore as DDS is based on a decoupled publish and subscribe paradigm it is best suited to applications in which one or more data sources (publishers) need to communicate information to one or more data users (subscribers), and where these sources and sinks are mostly decoupled.

Many applications have requirements to distribute both processing and data, especially within a SOA environment, and have often adopted the use of both CORBA and DDS. A significant advantage of both CORBA and DDS over alternate technologies is their ability to support highly heterogeneous and scalable systems. This makes them well suited for large, multi-vendor projects and for integrating applications that must run on diverse hardware, ranging from servers to embedded systems. Some of the capabilities shared by both DDS and CORBA include:

- Broad operating system support
- Broad hardware support, from enterprise through to embedded systems
- Integrated real time capabilities
- Support for multiple programming languages

As a result, both standards are part of the United States (US) Navy's Open Architecture Common Operating Environment (OACE) platform [Naval Surface Warfare Center Dahlgren Division 2004]. According to the US Department of Defense (DoD) Open Systems Joint Task Force and the US Navy Open Architecture initiative, an open systems

approach to architecture is "an integrated business and technical strategy that employs a modular design and, where appropriate, defines key interfaces using widely supported, consensus based standards that are published and maintained by a recognized industry standards organization" [Strei 2004b, p. 13]. The concepts of open architectures and open systems are enablers for the following benefits:

- Lower life cycle cost for weapon systems
- Better performing systems with greater interoperability
- Technology transparency for rapid upgrades
- Improved interoperability for joint warfighting
- Closer cooperation between commercial and military electronics industries

Figure 6 shows the US Navy's open architecture strategy. Level 3 OACE compliance is now a common requirement for most legacy system upgrades in the US today. The key to this compliance is standards based middleware, operating systems and mainstream COTS hardware. The major benefit of this imperative is to isolate applications from changes in computing technology and practice. The underlying hardware, networking, and operating systems can be replaced by newer, higher performing products and capabilities without affecting domain unique applications.

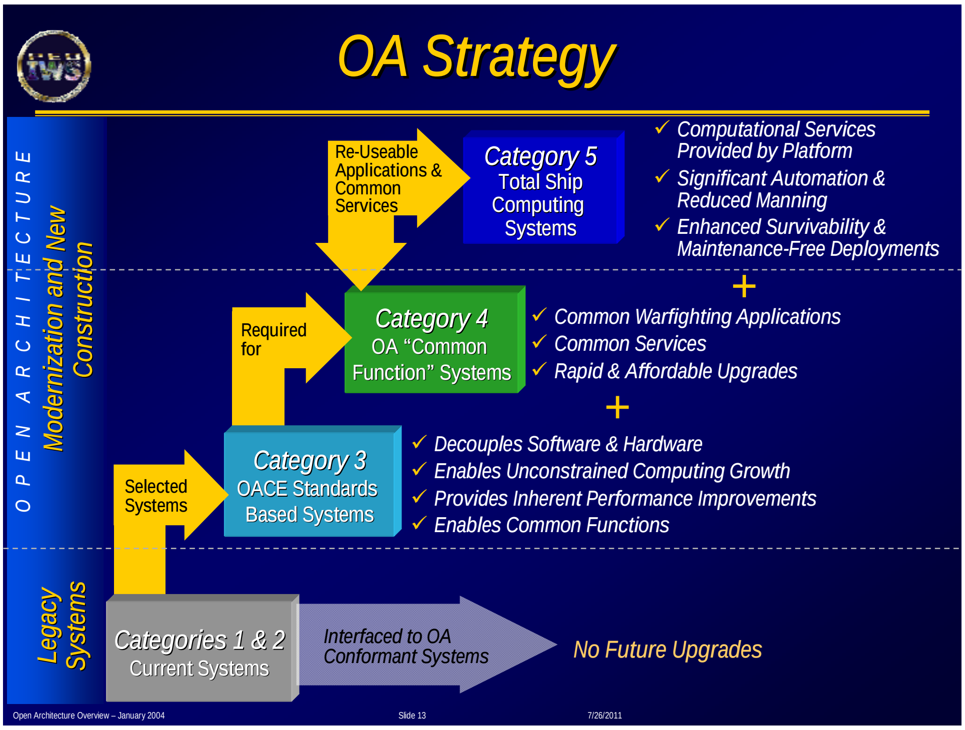


Figure 6: Open architecture adoption strategy overview [Strei 2004a, slide 13]

OACE defines both common infrastructure standards and component based design principles that that are combined to support the development of the common technology base. The set of open standards chosen for the OACE are illustrated in Figure 7 and include the following areas:

- Physical media – Telecommunications Industry Association (TIA)
- Networks and protocols – Internet Engineering Task Force (IETF)
- Operating systems – Institute of Electrical and Electronics Engineers (IEEE) POSIX
- Distribution middleware – OMG
- ADA and SQL languages – International Organization for Standardization (ISO)
- C++ language – American National Standards Institute (ANSI)
- Java programming language, infrastructure, Java Data Objects (JDO), Java Data-Base Connectivity (JDBC) – Java Community Process

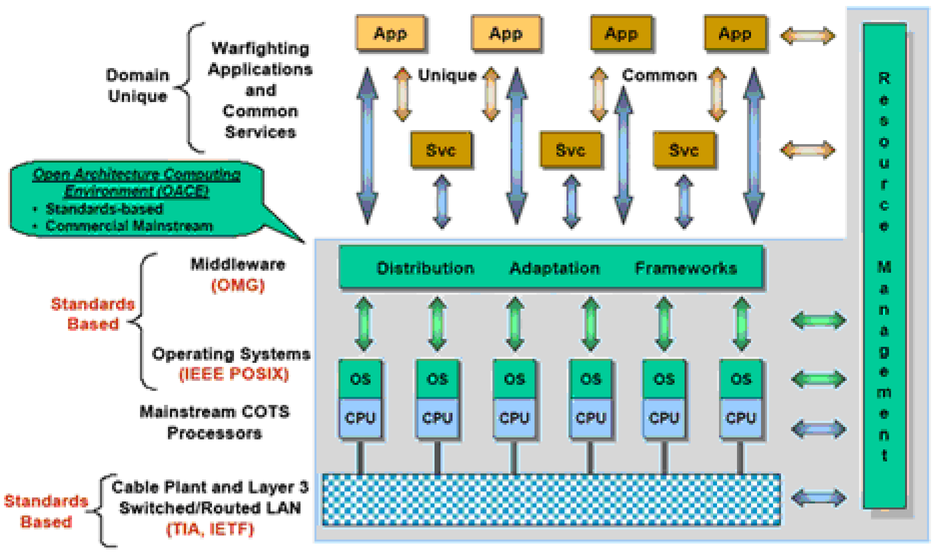


Figure 7: OACE design [Strei 2003, slide 10]

These same set of collaborating standards and guiding principals underpin the technology of NW-D10. The key objective of adopting these standards within a modular open systems approach to the architecture of military systems is to reduce the cost to develop, maintain and evolve these systems. An open architecture composed of these standards can have significant benefits in terms of preventing architectural obsolescence and enabling changes and upgrades over the lifetime of systems.

With the advent of high performance, COTS based, QoS enabled capabilities such as CORBA, DDS and CCM, mission critical military systems now have access to distributed middleware technologies that are based on open standards that can fully support the functional, non functional QoS and platform requirements of these systems. NW-D10

illustrated how both CORBA and DDS middleware technology standards can be used to integrate components at a node, system-of-nodes and systems-of-systems levels. Furthermore these technologies can potentially support the broadest spectrum of system requirements from small form factor real time embedded to large scale, providing a highly efficient, standards compliant and fully interoperable range of solutions at all levels within the system architecture.

It is the easy access to information that ranks as a central goal of the military's networked future. The US DoD's vision, called the Global Information Grid (GIG), calls for connecting systems across the globe. These systems range from enterprise business servers to battlefield tactical systems. The challenge is to get the right data to the right place at the right time. The enterprise portion of the GIG relies on commercial standards like Web Services and real time CORBA. However enterprise technologies like Web Services cannot address most of the real time requirements of tactical systems. These real time systems need targeted technology standards. The Net Warrior initiative has demonstrated through the NW-D10 event the real possibility of bridging these technological domains through standardised distributed component based technologies. This bridging capability, coupled with the plug-and-play nature inherent in distributed component based systems, means information residing within an enterprise domain can be merged with information resident in the tactical domain and vice versa. This merged information tends to form a continuous flow which can satisfy the warfighter's need to have pertinent and timely intelligence information contextually available in real time within a tactical situation. This continuous flow of information in turn enables a significantly enhanced decision capability to the war effectors or 'power-to-the-edge'.

Figure 8 depicts the Net Warrior reference architecture that supports these distributed component systems. The base layer represents the platform environment of operating system and services that run on top of hardware. The communications layer sits above the platform environment and provides standard transport and protocol support. Above the communications layer is the middleware environment that provides operating system abstraction and distribution standards and services to support networked components and systems. Specific domain environments at the top level provide a framework layer to host applications. Components are developed and deployed on top of the domain environment by extending the framework and may interact directly with the middleware environment.

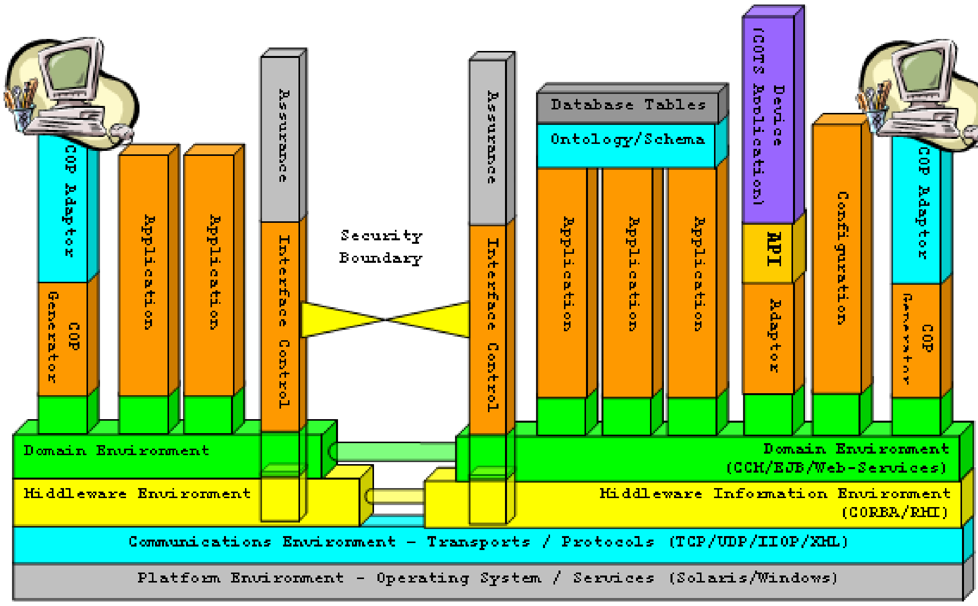


Figure 8: Net Warrior reference architecture

3.1.1 Distributed Object Computing

CORBA [Object Management Group 2008] is a mature open DOC infrastructure standardised by the OMG. CORBA automates many common network programming tasks such as: object registration, location and activation; request demultiplexing; framing and error handling; parameter marshalling and demarshalling; and operation dispatching.

A CORBA based system is a collection of objects that isolates the requestors of services (clients) from the providers of services (servants) by a well defined encapsulating interface. It is important to note that CORBA objects differ from typical programming objects as they can be:

- run on any platform;
- located anywhere on the network; and
- written in any computer language that has an Interface Description Language (IDL) mapping.

The OMG's Object Management Architecture (OMA) tries to define the various high level facilities that are necessary for distributed object oriented computing. The core of the OMA is the Object Request Broker (ORB), a mechanism that provides object location transparency, communication, and activation. The CORBA specification is based on the OMA and provides a description of the interfaces and facilities that must be provided by compliant ORB implementations.

CORBA is composed of five major components: ORB, IDL, Dynamic Invocation Interface (DII), Interface Repositories (IR) and object adapters. CORBA is therefore only a specification and must be implemented in software by a vendor. At its heart is the ORB, which is responsible for all the mechanisms required to:

- Find object implementation for requests,
- Prepare object implementation to receive requests, and
- Formulate and communicate the data making up requests.

Figure 9 illustrates the primary components in the OMG reference model architecture and the invocation control flows between distributed client and servant entities. The client is the entity that wishes to perform an operation on the object and the object implementation is the actual code and data that implements the object. Both the client and the object implementation are isolated from the ORB by an IDL interface. All requests are managed by the ORB. This means that every invocation (whether it is local or remote) of a CORBA object is passed to an ORB. In the case of a remote invocation, the request is further passed from the ORB of the client to the ORB of the servant object implementation through underlying communications transport mechanisms.

Since there is more than one CORBA implementation, and these implementations are possibly from different vendors it is important to ensure that these will also interoperate. It is however true that this interoperability of different ORB implementations has not always been guaranteed, especially prior to version 2 of the specification. This situation has now been rectified through further mandatory specification which now requires compliant ORB implementations to include both the General Inter-ORB Protocol (GIOP) and the Internet Inter-ORB Protocol (IIOP). The purpose of mandating these protocols is to ensure that clients are able to reliably communicate with servants written for different ORBs from different vendors and possibly even for different computer machine architectures and languages.

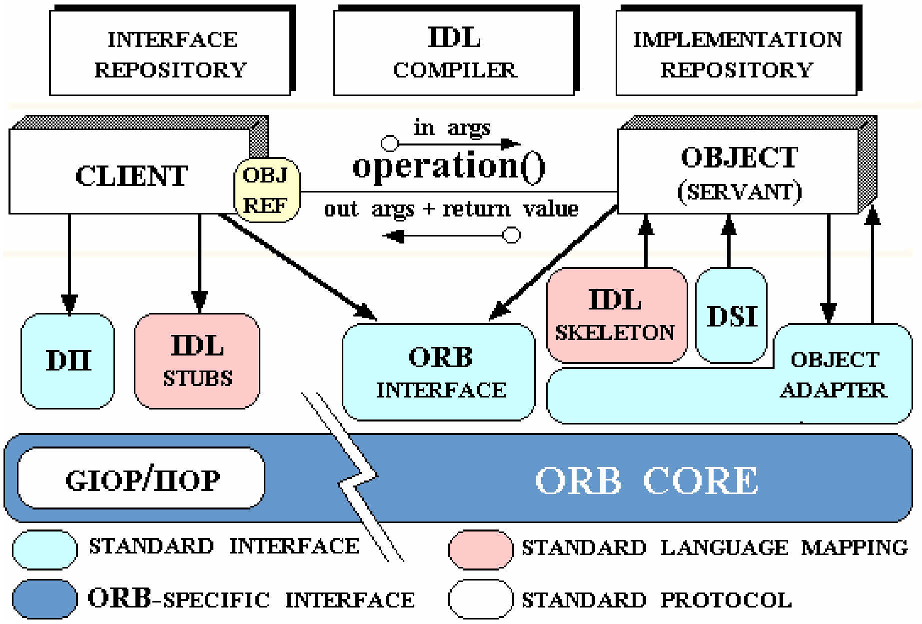


Figure 9: OMG CORBA reference architecture [Schmidt 2006]

As with the traditional and somewhat simpler remote method invocation constructs, CORBA objects are specified through abstract interfaces. In essence these interfaces form a contract between a client and a servant implementation. The CORBA specification further requires that these interfaces are specified in IDL. The IDL of a servant object interface also serves to define the object type (*typecode*). These IDL defined interfaces consist of a set of named operations and the set of parameters to those operations coupled with how they should also be invoked. However, IDL does not provide any implementation capabilities for the operations, nor is it a programming language despite its syntax being lexically similar to that of C++ and Java. It is from these IDL definitions that the CORBA object interfaces are mapped into different programming languages through *stubs* and abstract *skeleton* constructs particular to a computer language. Ultimately, it is through these mappings that a servant implementation is invoked by an ORB. It is also the same mechanism that allows for a client and a servant object implementation to use any computer programming language that has mappings from the originating IDL definition.

While CORBA allows for a sophisticated DOC environment, this alone is insufficient to satisfy a SOA paradigm. CORBA simply provides the necessary mechanisms and structure that allows for the construction of collaborative services of a SOA based capability. A SOA based system is concerned more about architecture and the delegation of responsibilities to components and component relationships than simply about the mechanisms and structure alone. It is however a precursor need of SOA that the mechanisms and structure of middleware technologies like CORBA are leveraged in the provision of system services and service collaboration.

3.1.2 Real Time Publish and Subscribe

In December 2004 the OMG formalised a standardised approach, *Data Distribution Service for Real-Time Systems Specification* [Object Management Group 2004], that addresses the need for a publish and subscribe capability within net centric environments. This DDS standard, which leveraged existing middleware network and communications capabilities, directly addresses distributed data centric informational needs common to most real time systems. Middleware is the software *fabric* that allows computer systems to opaquely exchange information over a network (Figure 10).

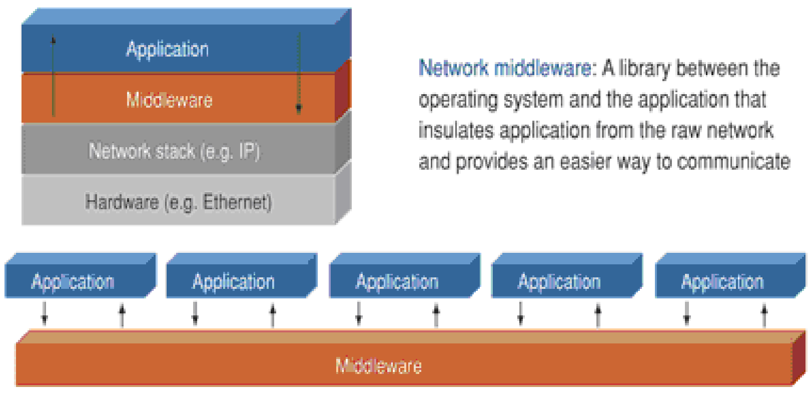


Figure 10: DDS contribution to the network middleware [Pardo-Castellote 2005]

DDS presents a publish and subscribe paradigm (Figure 11) that lets application nodes communicate by publishing information they have and subscribing to information they need in an asynchronous, decoupled manner. DDS also manages automatic discovery, reliability and redundancy over otherwise unreliable networks.

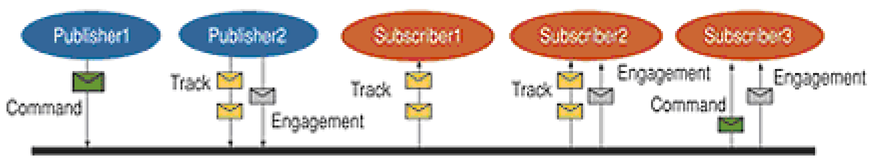


Figure 11: DDS topics for identification of data flows [Pardo-Castellote 2005]

The key to the power of DDS is its ability to flexibly but precisely specify performance requirements between all of the different parts of a system. DDS achieves this power through the use of QoS parameters that form the basis of contracts between publishers and subscribers (Figure 12). These QoS parameters prescribe exactly how information should flow between distributed nodes. It is these QoS parameters that provide applications with the performance guarantees and resource controls required by real time systems, while also preserving the modularity, scalability and robustness inherent in such an anonymous publish and subscribe model.

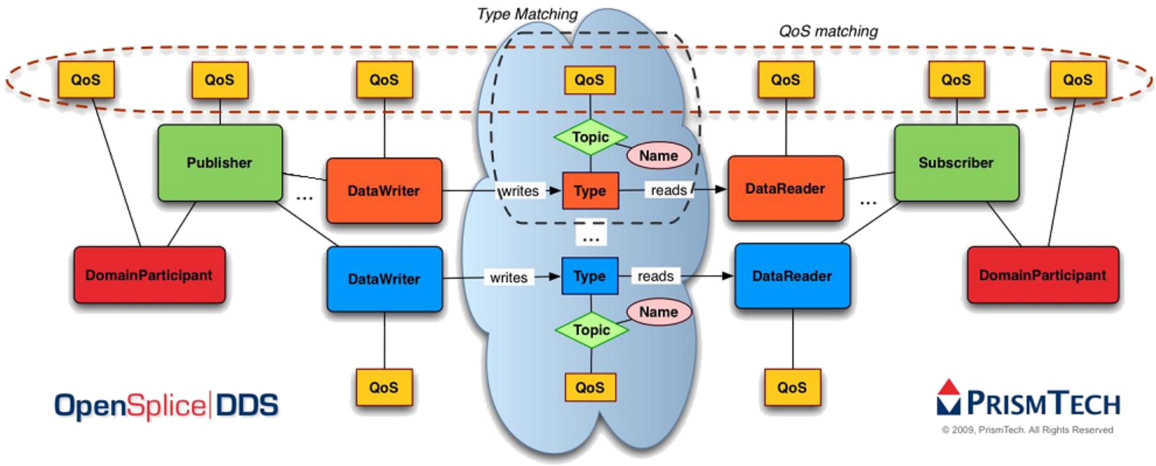


Figure 12: DDS entities and interfaces [Corsaro 2009, slide 12]

DDS enables applications to use a simple programming model when dealing with distributed data centric applications. Rather than developing custom event or messaging schemes, or creating wrapper CORBA objects to access data remotely, the application can identify the data as a *topic* it wishes to read and write using a name, and then through the use of the application programming interface (API), directly read and write that topic data.

The DDS publish and subscribe model connects anonymous information publishers (writers) with information subscribers (readers) within the context of a distributed application composed of separate component processes called *participants*. These processes are members of a logical domain, are run independently and commonly reside on separate computers. A participant may simultaneously publish (write) and subscribe (read) data flows identified through a shared but domain unique topic name. Operationally this can be pictured as a logical domain data bus. The data model allows the developer or system integrator to essentially ignore the complexity of the data flow itself and rely on each participant reading the data it needs from the bus.

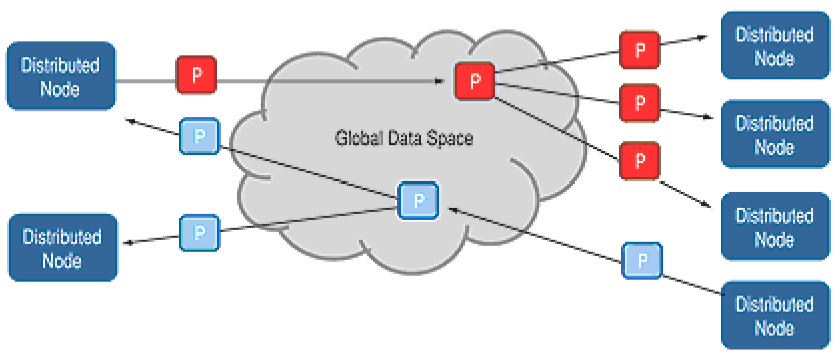


Figure 13: DDS realisation of a global data space [Pardo-Castellote 2005]

Participants using DDS can read and write data efficiently and naturally, with the underlying middleware distributing the data so that each reading participant can access the most current values. In effect, the service creates a *global data space* (Figure 13) that any eligible participant can read and write. It also creates a virtual name space called a *domain* that allows participants to find and share objects. In reality the data does not reside in any one computer or process, but rather it lives in the local caches of all the applications that have an interest in it. DDS also provides a type of state propagation model where nodes can treat data structures like distributed shared memory structures, with values being efficiently updated only when they change. There are also facilities in DDS that ensure coherent and ordered state integrity for these types of updates.

DDS is designed to automatically discover publishers and subscribers for each topic and autonomously establish data flows between them as qualified through QoS parameters. That makes it well suited for tactical and wireless environments where dynamic configuration changes are common. Unlike other distributed client/server technologies (e.g. CORBA), DDS does not rely on centralised repositories, specialised nodes or servers and is therefore highly resilient to partial failures in the network. At a fundamental level, DDS is designed to work over unreliable transports such as UDP, multicast or wireless networks, and tolerates brittle ad hoc connections. Efficient, direct, peer-to-peer communications, or even multicasting, are used to implement every part of the model.

DDS allows for fine control over QoS on a *per-data-flow* basis. Each publisher and subscriber pair can establish an independent QoS contract agreement. This aspect, which is unique to DDS, enables application designs that easily support extremely complex and flexible data flow requirements. QoS parameters control virtually every aspect of the DDS model even through to the underlying communications mechanisms. Most QoS parameters are implemented as a contract between publishers and subscribers; publishers offer and subscribers request levels of service. The middleware is responsible for determining if the offer can satisfy the request, thereby establishing the communication or alternatively indicating the incompatibility. Ensuring that participants meet their level-of-service contract guarantees predictability in operations necessary for real time systems.

DDS borrows much from its CORBA heritage including the standardised approach to format conversions across operating systems, processor architectures and programming languages. Furthermore, these characteristics make it ideally suited for the heterogeneous dynamic environments of the military tactical edge and especially to those introduced by the increased use of wireless mobile computing devices (such as smartphones and tablets).

Despite its relatively recent standardisation by the OMG the technology is well proven. The DDS standard unifies some of the best practices present in successfully deployed real time data distribution middleware. Since the finalisation of the DDS standard it has gained broad adoption. It has now been mandated as the data distribution middleware infrastructure for future programs by the US Navy OA and has already seen early adoption by many other military programs (e.g. AEGIS) [Strei 2004a].

3.1.3 Service Oriented Architecture

A SOA is an *architecture*. In itself it does not dictate the use of specific technologies, methodologies or environments. As an architecture its primary concern is the orchestration and organisational aspects of its component services rather than their implementation. However, a SOA does imply that its services are isolatable, deployable and that a communications path exists and is usable for interactions. A service in the context of an architecture therefore is as a constituent element of a SOA specifically tasked with the provision of a defined and bounded capability independent of any specific implementation or use. It is often only its context of use within a larger collection of other services that gives any distinct service a notion of function.

A service supports a formal interface *façade* which is most often both publically discoverable and accessible. It is only through this interface that the capabilities of a service can be invoked, with implementation specifics remaining veiled behind the façade. This veiling attribute is critically important to the organisational aspect of all services supported under a SOA, and allows services to easily become commoditised and replaceable.

Underpinning the SOA paradigm is the supporting infrastructure often termed collectively as *middleware*. This middleware, predominately based on open standards, is a collection of layered abstract frameworks and software pattern implementations. The term *middleware* is often incorrectly associated with product branding (i.e. DDS middleware); however it would be more correct to describe it as the totality of the capabilities of the infrastructure used exclusively in support of the componentisation of services and their interactions. This description therefore necessarily excludes those elements related to the platform, environment or computer language (i.e. operating system, virtual machine, POSIX etc).

A number of recognised middleware standards have emerged in the last 15 years with some waning to relative obscurity soon after their introduction. However the OMG, being a large consortia of hundreds of organisations, has standardised over the past decades most of the real time middleware standards of CORBA, CCM, DDS and other related companion standards in use today. These OMG standards have seen significant adoption by industry, predominately in real time control environments (e.g. medical and weapons). Furthermore, IBM introduced the Web Services product suite in 2004 to addresses the emerging enterprise environments of a mostly connected and relatively higher bandwidth Internet environment.

3.2 Middleware Adapters

Adapters are an implementation of the *adapter pattern* [Gamma et al. 1995], which is a model of behaviour that normalises two incompatible environments so that information and control flows can bridge this separation. Adapters generally have three distinct functional elements. The first element integrates the adapter to the infrastructure environment, typically a middleware platform, in accordance with a data model and other environmental conventions. These conventions are typically enforced through framework programming paradigms. The second element is a connector which integrates the adapter

with the other environment (usually an external device or system). Similarly, this is often achieved through the use of a framework or API. The third element incorporates an algorithm to perform any conversions, transforms or state synchronisation [Sioutis et al. 2008]. The adapters described in the following sections commonly utilise DDS for data transfer and are used for legacy system integration, cross domain bridging and visualisation. Other adapters can easily be incorporated in the future as the relevant patterns and development methodology are now well understood and can be used for rapid development.

3.2.1 WIRE Adapter

The primary external interface to the WIRE is though a Link 16 data link interface. However, due to technical limitations this interface was not working for the NW-D10 demonstration. In order to overcome this limitation a specialised adapter was developed that integrated with the Wedgetail software using compatible middleware technologies. The WIRE adapter was able to transfer track data to and from the Wedgetail mission system and DDS. Due to the chosen integration method with the WIRE this adapter is stateful. This means that transfer of data between the WIRE and DDS is not a simple matter of translation. The adapter needs to maintain an internal state of which tracks have been added, updated and deleted on each side as well as the frequency of the updates. There were no performance bottlenecks in this implementation so the adapter was able to utilise default DDS QoS settings.

3.2.2 XML Adapter

The XML adapter generates an XML stream with a simple schema that can be translated as needed by any external generic process. The output of the XML adapter is piped through a configurable UDP socket. The transform simply converts DDS tracks to the XML format.

3.2.3 Tactical Display Adapter

The Tactical Display Framework (TDF)[3] is a geographic information system visualisation application developed by Solipsys. Due to its customisable interface and extensibility, the TDF has been widely used within ADO programs to provide a situational air picture. The TDF is used in AOD to visualise tracks. This adapter transforms DDS data into proprietary formatted binary data as required by the TDF (e.g. converting coordinate systems and units appropriately).

3.2.4 Web Services Adapter

The Web Services adapter was utilised for integration with the ISRD ADIIB. This was achieved through simple translation of DDS track data to an equivalent Web Services description language schema and making a remote web request to an external server. Web Service technology is designed for integration with enterprise level systems running in

[3] More information on the TDF can be found on the Raytheon Solypsis website <http://www.solipsys.com>.

data centres with powerful servers and high bandwidth connections. After initial implementation it was quickly found that Web Services middleware, although highly flexible, executes much slower than CORBA and DDS middleware with equivalent functionality. This is not an issue when such software is executed in powerful servers but presents a bottleneck when executing in much more constrained military mission system computing architectures.

This adapter was receiving a great deal of data traffic from DDS and was not able to push it through Web Services quickly enough resulting in buffer overrun problems. It became evident that it was impossible to forward all DDS traffic over to the Web Services domain. Simply put, this adapter was required to filter the data. The filtering was achieved through heavy use of the DDS QoS capabilities. First, the DDS *reliability* QoS was set to BEST_EFFORT. This configured DDS to not guarantee delivery of data. This effectively removed the bottleneck from the rest of the DDS network as the adapter simply drops data when it reaches capacity. This was not the full solution as it was not possible to control which data was dropped. The additional utilisation of a DDS *data key* and *time based filter* (TBF) QoS solved this problem. These two settings allow DDS to filter data based on time and a specific field in the data, in this case a unique track identification number. For example, setting the TBF to 30 seconds configured DDS to only forward updates older than 30 seconds for each particular track. From the Web Services environment this appeared like all tracks were simply updating once per 30 seconds, which was a fair compromise.

3.3 Middleware Gateways

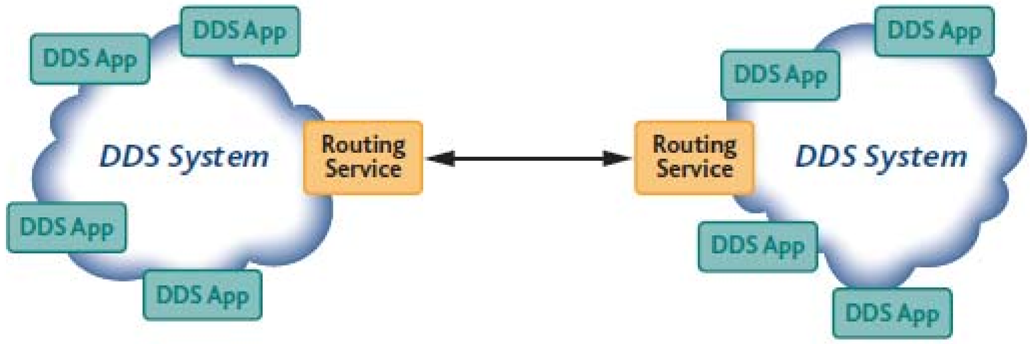
Middleware gateways are systems that are specifically designed for system integration. Their main task is to provide a means for connectivity as well any necessary translation of data.

3.3.1 Asynchronous Messaging Gateways

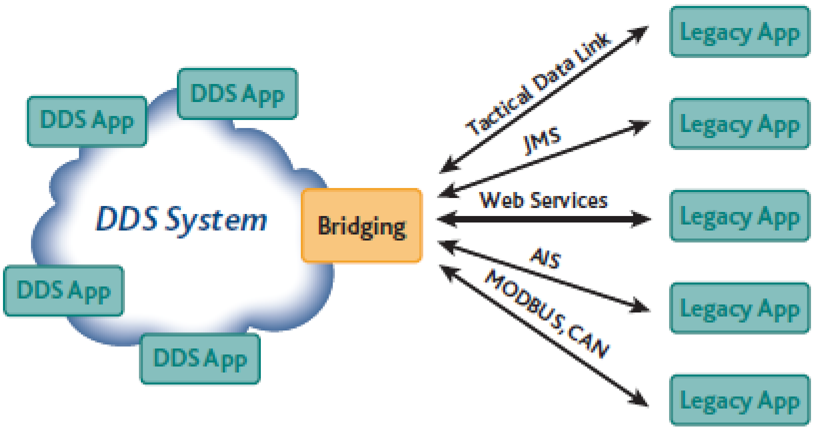
Asynchronous gateways are used to forward or translate data or messages between different domains. They are called asynchronous because they do not impose any synchronisation constraints between connected applications. They are also typically decoupled, which means senders and receivers do not rely on specific connections between them and can come and go as needed without impacting other systems.

The RTI routing service is an example of an asynchronous messaging gateway that can be used to forward and transform data. The forwarding or transforming of data is done in parallel to applications. Applications do not have to accommodate differences in data types or natively support different transports. Figure 14a illustrates the routing service connecting different DDS systems. The simplest example of this configuration would be connecting across DDS domains operating on the same network with the same data types. More complex scenarios would involve routing across different networks and translating into different data types.

The routing service can be extended by implementing custom connections in order to integrate with legacy systems. Figure 14b illustrates example legacy connections. This functionality is marketed as the "Adapter SDK" and is provided with an additional cost to the standard routing service.



(a) Domain forwarding and/or translation



(b) Transport bridging

Figure 14: RTI routing service typical usage examples [Real Time Innovations 2011]

The DSTO Java adapter framework was developed as a learning tool while researching asynchronous gateways. It utilises a central adapter object that supports a given data type. A number of connectors supporting different protocols can then be added to the adapter. When a connector utilises a different data type it needs to be added through a transform object that performs the data conversion. This framework was designed and implemented after identifying common patterns in the adapter implementations described in Section 3.2. Rather than having multiple separate adapter applications, many have since been refactored into connectors plugging into a larger adapter framework.

Figure 15 illustrates how the Java adapter framework works. Line 1 instantiates an adapter object which forwards objects of `TrackDetails`. It internally utilises a thread pool for handling multiple connectors in parallel. Lines 3 to 5 instantiate and add a connector for sending `TrackDetails` objects using the UDP protocol. This connector marshals data objects using the Java built in serialisation mechanism. Lines 7 to 10 instantiate another

UDP connector but it uses a `String` data type instead. This means that it cannot directly send `TrackDetails` objects and needs a transform to convert to and from `String` objects. The `TD2XML_Transform` performs this function by printing and parsing XML formatted `Strings` with the information contained in the `TrackDetails` objects.

The following two connectors are used to interoperate with DDS networks from two different vendors. They require two DDS topics for instantiation, one for sending and one for receiving. The KML connector (lines 11 to 13) utilises a HTTP service that can be used with Google Earth clients for geographic presentation.

```
1  Adapter<TrackDetails> adapter = new Adapter<TrackDetails>(3,10);
2
3  UDPConnector<TrackDetails> udp = new UDPConnector<TrackDetails>();
4  udp.connect(new UDPAddress("localhost:2000", 1000));
5  adapter.add(udp);
6
7  UDPConnector<String> udp_xml = new UDPConnector<String>();
8  udp_xml.connect(new UDPAddress("localhost:22224", 22223));
9  adapter.add(new TD2XML_Transform(udp_xml));
10
11 RTI_TDConnector rti = new RTI_TDConnector(0);
12 rti.connect(new DDS_Topics("RTI_OUT","RTI_IN"));
13 adapter.add(new Transform_TrackDetails_RTI_TD(rti));
14
15 OpenDDS_TDConnector odds = new RTI_TDConnector("0");
16 odds.connect(new DDS_Topics("ODDS_OUT","ODDS_IN"));
17 adapter.add(new Transform_TrackDetails_ODDS_TD(odds));
18
19 KMLConnector kml = new KMLConnector();
20 kml.connect(new KMLAddress(80, "C:\\http_files"));
21 adapter.add(kml);
22
```

Figure 15: DSTO Java adapter framework example usage

3.3.2 Synchronous Control Gateways

Synchronous gateways on the other hand are more complex because they can be used to forward method calls (i.e. control signals) with return values. They are called synchronous because they block the calling application until a response is received from the forwarded request. Synchronous gateways are required when clients and servants are located on different networks (or separated by a firewall) and/or use different transports. When more that one transport is available the gateway can automatically switch between them based on a given policy. Additionally, synchronous gateways can be used to enforce policies for connectivity (e.g. security).

The primary method used to implement synchronous gateways is through a Dynamic Skeleton/Invocation Interface (DSI/DII). The DSI is used to receive arbitrary requests from a CORBA system and the DII allows CORBA invocations to be made dynamically. The DSI/DII combination allows a gateway to accept invocations on any specified set of interfaces and pass them to another system.

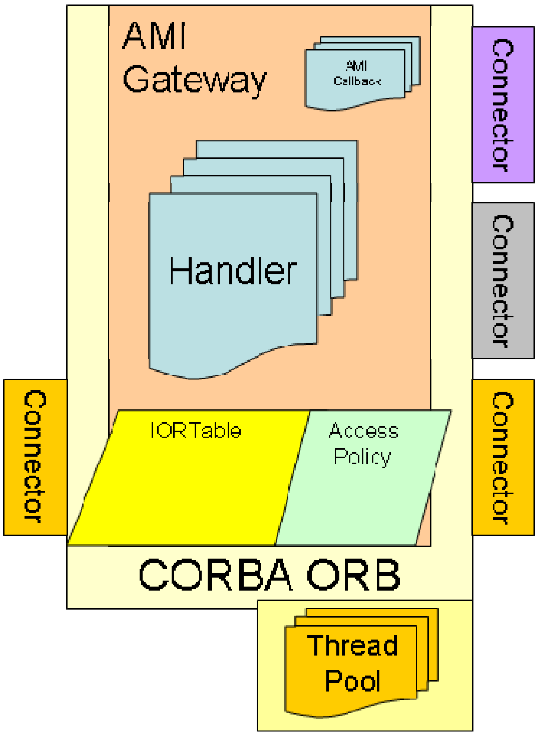


Figure 16: DSTO AMI gateway

The DSTO asynchronous message invocation (AMI) gateway is used to provide a front end interface for a large number of CORBA objects. CORBA servants need to register with the gateway giving a unique name identifier. The gateway maintains a mapping between the identifier and the servant interoperable object reference. This registration also serves to validate what is accessible (further work would institute credential validation through incoming request interceptor logic). When an invocation is received on that identifier (through DSI), the gateway generates a request element and associated reply handler through the registered servant handler and then forwards it to the associated servant (through DII). Subsequently the reply handler receives any related result or exceptions along with any optional arguments and returns these to the calling client which results in it becoming unblocked.

The gateway uses AMI (Figure 16) because it allows one to decouple requests from their replies. In other words it becomes possible to send a request and later get a callback when the result returns from the servant. The gateway uses a thread pool to concurrently dispatch different threads for the following tasks: a) registering new servants, b) handling new client invocations, c) forwarding requests to servants, and d) handling callbacks and unblocking clients with the results. This architecture allows the DSTO AMI Gateway to be very efficient, handling multiple requests in parallel with minimal resource consumption.

3.4 Tactical Data Links

3.4.1 Tactical Data Link Gateways

Laboratory-based experiments allow DSTO to gain a better understanding of the potential benefits of TDL gateways and future technologies and how ADF platforms fit into the future GIG or Australian Regional Information Grid.

The cross environment TDL integration demonstrated in NW-D10 was achieved through the utilisation of a COTS TDL gateway. The particular gateway employed in the ASCEL was the Gateway Manager (GM), however there are many other similar systems available in the market. Differences between different vendors include but are not limited to: a) TDL types and versions, b) physical interfaces, c) protocols (e.g. terminal emulation), d) scenario generation, e) recording and playback capability and f) provision of an API. One important feature of a TDL gateway that is often neglected is its *certification*. That is, gateways that have been officially certified to work with particular types of TDL, versions and interfaces. Typically, TDL gateways will not be used in production TDL networks unless they have the appropriate certification.

3.4.2 Network Management Systems

JP2089 is established in the Defence Capability Plan to develop Defence's tactical information exchange domain. JP 2089 Phase 2A includes the Initial Common Support Infrastructure component, which is intended to further the development of Defence's Joint Interface Control capability through the provision of sufficient systems to enable 'proof of capability' and analysis-of-alternatives activities to be undertaken on a multi-TDL network management system (NMS).

DSTO utilises the ASCEL to conduct research in support of JP2089, which includes involvement in Net Warrior events. In NW-D10 the ARMS NMS was used to monitor a Link 16 network that included both scripted and live participants. The result of these activities will contribute to the functional performance specification for Defence's mature TDL network support infrastructure, which will be delivered by JP 2089 Phase 3 over 2011 to 2013.

3.4.3 Data Link to Middleware Integration

During the NW-D10 event, information appearing in the TDL network was forwarded across to the middleware infrastructure. This was done through the use of a TDL adapter which bridged TDL traffic to DDS. The TDL adapter utilised another instance of the Rosetta application. Rosetta maintains an internal database of tracks received via TDL that can be manipulated by an API to both send and receive TDL data.

The particular version of Rosetta used in NW-D10 required querying its database for the entire list of tracks. The adapter iterated through this list and published the data over DDS. The disadvantage of this approach was that the Rosetta database was queried once every few seconds and all tracks were published whether they had been externally updated or

not. This meant there was a large burst of DDS traffic every time this adapter cycled through its database. Later versions of Rosetta support real time forwarding to predefined XML schemas. This adapter has since been upgraded to utilise the XML feature and effectively provides real time bidirectional DDS to Link 16 integration without the need to use the Rosetta API.

4 Outcomes of the NW-D10 Event

NW-D10 demonstrated the utility of interconnected informational technologies resident within a network of four DSTO battlelabs. Distributed exercises highlight the challenges of creating functional and *seamless* systems of systems across multiple domains. NW-D10 fostered a small community of cross disciplinary applied research that is of operational significance. It demonstrated a way forward for enhanced situation pictures for Regional Operations Centres (ROC), Wedgetail and other tactical and command and control nodes and generated a multitude of ideas for future collaborative work.

The NW-D10 event involved the integration of a number of technologies and methodologies, including SOA, CORBA, DDS, Web Services, TDLs (Link 16), component based systems and visualisation tools. Seamless information interoperability between disparate systems was demonstrated. Achieving this information interoperability was a non trivial research task that was lead by AOD. It required the definition of a common data ontology for information translation and the development of adaptors and bridges or gateways used to integrate the flow of information between the systems. With a common high level goal, each of the participating divisions focused on their enabling research programs and client needs. In summary the NW-D10 outcomes for AOD were:

- deeper understanding of tactical NCW integration challenges through a ‘learn by doing’ approach;
- enabling research that developed the understanding of the application of SOA methodologies to the domain to manage complexity, risk and to bridge domains;
- demonstration to ADO stakeholders of adaptation and integration technologies for seamless information interoperability between disparate tactical and enterprise systems; and
- successful collaboration across multiple divisions to develop and demonstrate NCW concepts using the Net Warrior battlelab network infrastructure.

4.1 Learn-by-doing Approach

Net centric systems are dynamic, large scale ‘systems of systems’. They must operate in heterogeneous and complex domains while constrained by stringent simultaneous QoS demands. In the military tactical domain the development and integration of such systems

is extremely difficult, and once in service they undergo continuous evolution and change making sustainment and growth a major challenge. The need for interoperability with legacy military and civilian systems compounds the complexity.

Traditionally, modelling and analysis has been used to understand and analyse the performance of such systems. However, by necessity, models are limited representations of the real world with abstract complexity. Modelling and analysis is particularly limited when applied to understanding integration risk in complex domains.

The NCW Roadmap [Department of Defence 2009] advocates a learn-by-doing approach, primarily directed to the human dimension. Net Warrior extends this approach to the network dimension for experimentation with the enabling communications infrastructure, information systems and common information services.

NW-D10 used high fidelity representations of real systems, including Wedgetail operational software, data link systems, and enterprise systems. This was essential to understand the complexity of the systems and their integration. In NW-D10, these systems were stimulated in the same way as they would in operations but in networked laboratory environments. This is a step beyond modelling and simulation in both complexity and fidelity.

Table 1 shows the system readiness levels (SRLs) as defined in the DSTO Technical Risk Assessment Handbook [Defence Science and Technology Organisation 2010]. The SRLs provide an assessment of the maturity of a system and its integration. Modelling and analysis can be used for evaluation of a system up to SRL 3. NW-D10 demonstrated stimulated systems in a simulated environment with limited external data feeds, and so can be considered to have evaluated and hence potentially mitigated risk at an SRL of between 5 and 6.

Table 1 System readiness levels. Source [Defence Science and Technology Organisation 2010]

System Readiness Description	Readiness Level
Basic principles observed and reported.	1
System concept and/or application formulated.	2
Analytical studies and experimentation on system elements.	3
Sub-system components integrated in a laboratory environment.	4
System tested in a simulated environment.	5
System demonstrated in a simulated operational environment, including interaction with simulations of external systems.	6
Demonstration of system prototype in an operational environment, including interaction with external systems.	7
System proven to work in the operational environment, including integration with external systems.	8
Application of the system under operational mission conditions.	9

Figure 17 shows a simplified view of the increasing complexity of risk mitigation through analysis, simulation, and stimulation through to operational systems. The Net Warrior learn-by-doing approach using high fidelity representations of real systems, as applied in NW-D10, has proven to provide a deeper understanding of tactical NCW integration issues and technical challenges than would have been possible through purely using modelling and analysis.

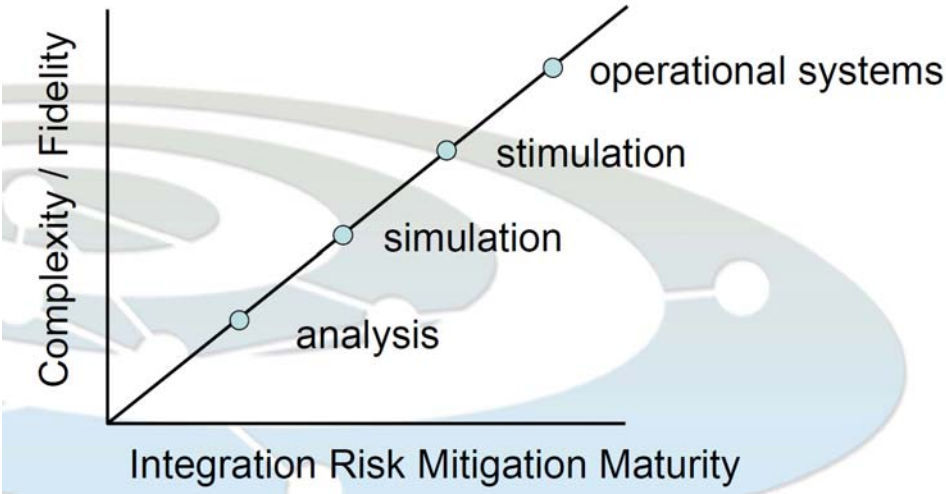


Figure 17: Experimentation with laboratory-based systems that are high fidelity representations containing both operational and simulated elements

4.2 Management of Complexity and Risk

A key advantage of the utilisation of SOA for integration is that it provides the means to manage complexity as more and more systems are connected together. When utilising a SOA approach, systems are not integrated 'horizontally'. This means that there is no specialised (and direct) connectivity between such systems. Instead, they are integrated 'vertically' through the utilisation of middleware. The job of the middleware is to transparently and efficiently connect all the systems together and achieve this through a set of layers (also called tiers) that conform to standards based interfaces and protocols. There are a number of advantages to this approach, for example:

- Systems need only be integrated into a layer that they can support. The middleware has the responsibility to transform data and control information into formats that other systems can understand.
- It is much easier and cheaper to introduce changes within large (composite) systems by simply plugging in to the middleware and joining the system workflow.
- Systems can easily be replaced (or upgraded) with no impact to peers as long as the original interfaces are supported.

- Middleware frameworks essentially provide partial applications and utilising them greatly reduces the implementation size and complexity, hence reducing development costs.

NW-D10 showcased the use of SOA for cross environment integration. The integrated systems were distributed across different buildings and had been developed by different teams for different environments, specifically air, command and control, and ISR. NW-D10 demonstrated that there were three 'information buses' required to achieve this, specifically:

- a legacy tactical data link bus,
- a real time publish/subscribe and synchronous control bus, and
- an enterprise information service bus.

Different systems were integrated into the information bus architecture they could support and became active members of that environment. Additionally, the three information buses were themselves bridged using custom middleware gateways. COTS middleware gateways do exist for this purpose but were not utilised in order for DSTO to better understand what is required at that level. This effectively achieved a seamless cross environment integration of systems that normally could not interoperate. NW-D10 demonstrated this concept through a mock homeland security scenario with an NCW flavour. The scenario was designed so that the mission commander would require information from all of the different environments. This information was made available on demand and presented as part of the commander's situational picture.

4.3 Client Feedback

The NW-D10 was repeated three times in September 2010 and attracted representation from many parts of the ADO, including CDG, CIOG, DMO, HQJOC, SRG, AFHQ and the DSTO senior executive. The demonstration was the first time the client community was exposed in detail to the Net Warrior program and approach.

There was significant hidden complexity in NW-D10 that was difficult to convey in the short demonstrations; however ADO stakeholders increased their awareness of technical issues with NCW integration and will be better informed when acquiring new capabilities and growing extant capabilities. Commander SRG commented that an awareness of the keys aspects of SOA technologies and networked high fidelity laboratory-based systems will be of use in future training systems and integration challenges. Experimentation with these enabling technologies and the integration of high fidelity laboratory-based systems with these technologies are an important component for enabling the ADF to become a net centric force. It was noted that NW-D10 was focused on air related capability but can easily be extended to other domains. Specific comments from attendees were that Net Warrior, as demonstrated through NW-D10, is:

- very relevant to the capability integration and tactical SOA questions considered by CDG, CIOG and DMO and can inform future enterprise SOA approaches;
- considering issues that are relevant to a number of current and future Defence Capability Plan activities and offers the potential to mitigate significant risk for these projects;
- well placed to explore SOA at the tactical edge;
- allowing Defence to use high fidelity laboratory-based systems containing operational elements to understand and manage the complexity of software and integration of systems in a NCW context; and
- aligned with the intent of a number of the NCW Integration and Implementation Strategy (NCWIIS) [Department of Defence 2010] initiatives.

In support of the NCWIIS, Net Warrior is positioned to inform validation and demonstration of the networked force design (supporting Imperative 3), conduct operational testing to evolve the networked force (supporting Imperative 6), and possibly inform future capability in-service through modelling and simulation (supporting Imperative 7).

Net Warrior uses Network Centric Operations Industry Consortium (NCOIC) architectural reference models, which are slightly different to those adopted by CIOG. This does not invalidate current work undertaken by DSTO but should be coordinated with CIOG and other key Defence players to optimise the approach adopted. The 'tactical SOA' approach may need to be harmonised with that currently being proposed by CIOG at an enterprise level.

Moving into the future, Net Warrior is well placed for human-in-the-loop experimentation to explore NCW human dimension issues. Net Warrior could be utilised to optimise force design and operation as other major NCW nodes such as the Air Warfare Destroyer (AWD) and Landing Helicopter Dock Ships. For example, this could be a cost effective way to explore how Wedgetail and the AWD should operate together in an ADF context.

To achieve this, closer engagement is required between CDG, CIOG and DSTO. As a direct result of stakeholder exposure through NW-D10, DSTO presented on SOA integration concepts at the CIOG Defence Enterprise Architecture Council and Defence Enterprise Architecture Working Group in November 2010.

The NW-D10 event fostered a community of cross disciplinary researchers who understand that the collaboration will result in outputs with operational benefits. It demonstrated ways forward for enhanced situation pictures for the ROC, Wedgetail and other tactical and command and control nodes and has generated a multitude of ideas for future collaborative work of value to clients.

4.4 Multi-Divisional Collaboration

Through collaboration between three DSTO divisions, NW-D10 demonstrated the utility of interconnected informational technologies resident within a network of four DSTO battlelabs: the WIRE, ASCEL, FOCAL and ISRAIL. This distributed demonstration highlighted:

- the challenges of creating functional and seamless systems of systems across multiple domains;
- organisational challenges that must be overcome for NCW integration across multiple disciplines and domains; and
- the difficulty in demonstrating to clients the underlying distributed technology in an operational context.

These outcomes could not have been achieved without the successful cross disciplinary collaborative effort. Cross fertilisation of ideas across DSTO was valuable and a multitude of ideas were raised for future collaborative work to develop and demonstrate NCW concepts under Net Warrior.

5 Future Net Warrior Events and Enabling Research

The AOD Net Warrior community is continuing to investigate the utility of applying modern distributed object computing technologies to the networked systems integration domain. Future planned events will encompass scenarios that will attempt to bind the enterprise domains of defence with the real time tactical aspects of the war fighter. Initially these events will predominately focus on the integration of testbeds that are representative of airborne mission systems contained on both the AEW&C and F-35 Lightning II platforms.

Further activities and events will enhance this capability with the introduction of commercially available mobile technologies that can be representative of a number of distinct network nodes that are both agile and self reliant. It is expected that such devices will act as both remote sensors and a capability to provide real time situational perspective to the end user.

In support of these objectives the AOD Net Warrior community has a burgeoning enabling research program where emerging technologies and capabilities can be evaluated within the context of an informational network of high fidelity laboratory-based systems containing operational elements. There are many emerging technologies that could address aspects of these systems of systems integration challenges. However, it is only those that potentially address the future integration needs of the ADF and planned Net

Warrior events and are currently considered by AOD as mature and stable technologies that are at this stage being investigated. These are:

- agent technologies as applied to DOC and SOA environments for automated decision support;
- mobile technologies with a tactical sensor suite and situational communication endpoint;
- CCM architectures for avionics mission systems integration; and
- distributed mission training to investigate the next generation training systems for the Royal Australian Air Force (RAAF).

Each of these areas of enabling research are briefly discussed below, and future Net Warrior events are targeted for showcasing elements of these technologies being applied to the networked environment of the ADF.

5.1 CORBA Component Model

CCM is a relatively recent addition to the family of CORBA definitions. It was introduced with CORBA version 3 and describes a standard application framework for CORBA components within a DOC environment. In consideration of the technological accomplishments of NW-D10 where much of these same technologies (CORBA/DDS) were used for integration, it is reasonable that such a standardised approach would serve as beneficial for future network integration effort.

5.2 Agent Technologies

Agents embody a software development paradigm that merges theories developed in artificial intelligence (AI) research combined with computer science. The power of agents comes from their intelligence and also their ability to communicate and share information. Current agent development methodologies and resulting frameworks have been developed from an AI perspective. They introduce a number of behavioural concepts (e.g. goals, beliefs) and provide support in the event processing, resource management and structure of their implementation. There is great emphasis placed on hiding the complexity of the underlying AI algorithms upon which the agents operate. However, agent systems are inherently distributed software systems and this brings significant implications in their application and, more importantly, their integration.

In the context of DOC, and specifically SOA, agents can be viewed as autonomous services with specialised algorithms utilised for intelligent behaviour. DOC middleware can provide the infrastructure upon which agents communicate with one another, as well as sense and act upon the environment. When developing agents it is possible to recognise and decompose the patterns of behaviour that agent frameworks implement. These behaviours can then be described using a combination of software design patterns. This

future work will explore the possibility of leveraging the power of both approaches through a framework that implements generic agent behaviours and algorithms with DOC middleware using well understood software design patterns. A developer can subsequently utilise the same middleware employed in their SOA systems and at the same time introduce agents with very little integration risk. [Sioutis & Dominish 2011] provides a detailed discussion of this research program.

5.3 Mobile Technology

Mobile computing devices, such as smartphones and tablets, are inexpensive and sophisticated electronic devices. Recent advances have included graphical displays, keyboards and other native sensor and networking capabilities. In the future it is possible that mobile technology could contribute to some of the tactical edge coordination activities of high value ADF assets (e.g. fighter, weapon and surveillance platforms). These devices might also be used by war fighters to access or provide critical operational information, such as intelligence, terrain descriptions, maps or asset descriptions.

In late 2010 research under Net Warrior was extended to investigate how mobile computing devices could be integrated into tactical SOA environments (e.g. with restricted bandwidth and unreliable links). The goals of this research program are to investigate:

- how mobile devices could be integrated into the Australian NCW environment;
- information interoperability between mobile devices and high value defence assets (initially represented by an airborne mission system testbed) to enable increased situational awareness; and
- the utility of mobile devices for air-surface integration.

A smartphone (specifically a 16 GB iPhone 4) is being used as a representative mobile device but this could easily be a smartphone or tablet from a different manufacturer and with a different operating system (e.g. Android or Windows Mobile). The initial work conducted has investigated how distributed object and publish/subscribe middleware can be incorporated into a smartphone to achieve information interoperability. The middleware technologies chosen are CORBA and DDS. These technologies are suited to low bandwidth tactical environments because utilisation of the underlying communication bearers can be tightly controlled. In the future, the focus will shift towards integrating the iPhone with an airborne mission system testbed through CORBA and DDS to provide a simple air picture. Further research activities could focus on end user interactive application development (i.e. chat) and incorporating camera and voice capabilities into future ISR applications. [Foster 2012] provides a detailed discussion of this research program.

5.4 Distributed Mission Training

The following systems are all being developed within AOD:

- The Air Defence Ground Environment SIMulator (ADGESIM) – a high fidelity, stimulated, real air defence mission system. ADGESIM is currently incorporating Link 16 and it will then be compliant with the proposed ADF corporate LVC (live, virtual and constructive) synthetic range interoperability model [Zalcman & Blacklock 2010].
- The Air Operations Simulation Centre (AOSC) Desktop Aircraft Cockpit Simulator (DACS) – an emulated F/A-18 human-in-the-loop system that is currently being developed with the same synthetic range interoperability model.
- The WIRE – a high fidelity, stimulated, real mission system. The WIRE is currently being enhanced to have a Synthetic Range Interoperability Model capability.

These three systems could be made LVC interoperable by being made compliant with the synthetic range interoperability model. This is already occurring for the ADGESIM and DACS systems, but would need to be done for the WIRE. By adding CGF, after action review, logging and other capabilities these interoperable systems could then be developed into an AOD air battle management mission training centre.

Initially this would involve providing the required LVC interoperability between the AOD ADGESIM, DACS and WIRE systems and investigating what CGF, after action review, logging and other capabilities could be added to the synthetic environment formed by these systems. This would require the development of a Net Warrior LVC interoperability strategy and a set of Net Warrior LVC interoperability standards [Zalcman et al. 2011]. These Net Warrior LVC interoperability standards would include the development of Net Warrior LVC common object models, common gateways and common federation agreements.

It is intended that the WIRE capability as extended under the Net Warrior Initiative and demonstrated under NW-D10 will be used within the context of Exercise Black Skies 2012. Black Skies is an exercise conducted by AOD every two years in support of the live air combat training exercise Pitch Black. Black Skies enables new simulation tools and training techniques to be evaluated and developed for future implementation within the RAAF. Therefore, Black Skies is the ideal vehicle for testing the distributed mission training concept (i.e. networked operational mission systems wrapped in stimulation environments).

5.5 Tactical Data Link Research

Another aspect of TDL research being conducted at DSTO is the derisking of TDL integration between systems being procured by the ADO. One example of such an activity involves the TDL integration of the WIRE and the 9LV VirtualShip testbed being developed in Maritime Operations Division (MOD) at DSTO. At this time, both of these

systems are being upgraded with a full TDL capability and the ASCEL will be utilised to verify their interoperability. Any issues identified by this testing will be raised with the respective projects in order to inform their resource planning and risk management strategies.

6 Conclusion

The transformation to an Australian net centric force requires a shift in the way systems are procured, built and used, so that information can flow through a changing network of heterogeneous nodes, each with its own information requirements. For example, aircraft, due to their mobility, have a continuously changing context of operation thus requiring dynamic connections to other network nodes. NW-D10 demonstrated the utility of such technological environments and infrastructure to conduct research into how information flows can be agile and adaptable within the dynamic and distributed environments of mission systems.

The technologies and methodologies presented in NW-D10 enable systems to grow in an evolutionary manner by adapting to emerging technologies. For example, by using common and open standards to define interfaces, existing technologies can more easily be integrated or replaced with newer ones. It is through the Net Warrior initiative that much of the capabilities implicit with these emerging open standard technologies can be evaluated within the context of existing systems usage and warfighter doctrine.

Crucial to the future work of the Net Warrior initiative is the specification and adoption of a set of guidelines for system architecture that encompasses aspects of the underlying open technology standards both in structural organisation and inter and intra system orchestration of workflows into a services paradigm (i.e. SOA).

NW-D10 served to show the utility of applying modern middleware technologies to integrate mission and combat systems into a seamless networked informational environment. Systems connected within and through this networked environment can become aware of critical information that can be relevant to their individual decision support functions. It is when these systems adapt to these new informational sources that the warfighter, who is becoming increasingly dependant on these systems, can gain a fundamental informational advantage over an adversary. Furthermore, as these systems may also harbour organic information which is often sourced from their own sensors this can aid in the situational understandings of the wider networked audience and the systems that are used in support of those understandings. NW-D10 also served to show that it is not just the enhanced awareness that is a critical factor in a networked force but also the capacity to improve the effectiveness of warfighters who become reliant on systems that adapt to best utilise these new informational sources in making timely and often critical decisions.

Net Warrior technology demonstrations provide a dynamic and evolving capability. Experimentation with enabling technologies and the integration of high fidelity laboratory-based systems containing operational elements with these technologies is an important component for enabling the ADF to become a net centric force. Results from this experimentation will enable Defence to be better informed when acquiring capabilities that interoperate with other systems.

7 References

Corsaro, A 2009, *OpenSplice DDS: a crash course*, PrismTech, viewed 15 August 2011, <http://www.prismtech.com/opensplice/resources/demand-webcasts/opensplice-dds-crash-course>.

Defence Materiel Organisation 2009, *Air 5077 Project Wedgetail Airborne Early Warning and Control Program*, Defence Materiel Organisation, Canberra, viewed 11 July 2011, <http://www.defence.gov.au/dmo/dceod/air5077/air5077.cfm>.

Defence Science and Technology Organisation 2010, *Technical Risk Assessment Handbook*, version 1.1, Defence Science and Technology Organisation, Canberra.

Department of Defence 2009, *NCW Roadmap 2009*, Capability Development Group, Department of Defence, Canberra.

Department of Defence 2010, *NCW Integration and Implementation Strategy 2010*, Capability Development Group, Department of Defence, Canberra.

Filippidis, A, Doan, T, & Tobin, B 2007, 'Net Warrior – DSTO battlelab interoperability', *Proceedings of the SimTecT 2007 simulation conference and exhibition*, Simulation Industry Association of Australia, Brisbane.

Filippidis, A & Gencarelli, M 2008, 'QoS of tactical information exchanges in Net Warrior', *Proceedings of the Land Warfare Conference*, Brisbane.

Foster, K 2012, 'Middleware interoperability in tactical defence environments: smartphone integration in the Net Warrior initiative', *Proceedings of the 16th International Conference on Knowledge Based & Intelligent Information & Engineering Systems (KES2012)*, Spain.

Foster, K, Iannos, A, Lawrie, G, Temple, P & Tobin, B 2007, *Exploring a net centric architecture using the Net Warrior AEW&C node*, DSTO-TR-2093, Defence Science and Technology Organisation, Edinburgh, South Australia.

Gamma E, Helm, R, Johnson, R & Vlissides, J 1995, *Design patterns: elements of reusable object-oriented software*, Addison Wesley.

Naval Surface Warfare Center Dahlgren Division 2004, *Open Architecture (OA) Computing Environment Design Guidance*, version 1.0.

Object Management Group 2000, Common Object Request Broker Architecture Specification, version 2.4.

Object Management Group 2004, Data Distribution Service for Real-Time Systems Specification, version 1.0.

Object Management Group 2007, Data Distribution Service for Real-Time Systems Specification, version 1.2.

Object Management Group 2008, Common Object Request Broker Architecture Specification, version 3.1.

Pardo-Castellote, G 2005, 'DDS spec outfits publish-subscribe technology for the GIG', *The Journal of Military Electronics & Computing*, viewed 15 August 2011, <http://www.cotsjournalonline.com/articles/view/100296>.

Real Time Innovations 2011, 'RTI Routing Service for DDS', Data Sheet.

Schmidt, D 2006, Overview of CORBA, viewed 16 August 2011, <http://www.cs.wustl.edu/~schmidt/corba-overview.html>.

Sioutis, C & Dominish, D 2011, 'Developing intelligent agents with data distribution middleware', *Proceedings of the KES third international conference on intelligent decision technologies (KES IDT 11)*, Athens.

Sioutis, C, Foster, K, Temple, P & Dominish, D 2008, 'Achieving information interoperability using data distribution middleware', *Proceedings of the military communications and information systems (MILCIS) conference*, Canberra.

Strei, T 2003, 'Open architecture overview', *Proceedings of the sixth annual National Defense Industrial Association system engineering conference*, National Defense Industrial Association, San Diego, California.

Strei, T 2004a, Open architecture: an enterprise approach to introducing open architectures into navy combat systems…and beyond, viewed 25 July 2011, <http://www.acq.osd.mil/osjtf/mspowerpoint/NOAbrf.ppt>.

Strei, T 2004b, 'Overview of naval open systems: the plans, the approach, the promise', *Proceedings of the sixteenth annual systems & software technology conference*, Salt Lake City, Utah.

Zalcman, L & Blacklock, J 2010, USAF distributed mission operation, an ADF synthetic range interoperability model and an AOD mission training centre capability concept demonstrator – what are they and why does the RAAF need them?, DSTO-TR-2463, Defence Science and Technology Organisation, Fishermans Bend, Victoria.

Zalcman, L, Blacklock, J, Filippidis, A & Lawrie, G 2006, 'The DSTO Net Warrior exercise – the ADGESIM contribution', *Proceedings of the SimTecT 2006 simulation conference and exhibition*, Simulation Industry Association of Australia, Melbourne, pp. 339-344.

Zalcman, L, Blacklock, J, Foster, K & Lawrie, G 2011, *An AOD LVC corporate interoperability standards development strategy*, DSTO-GD-0645, Defence Science and Technology Organisation, Fishermans Bend, Victoria.

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA	1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)

2. TITLE	3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)
Net Warrior D10 Technology Report: Airborne Early Warning and Control (AEW&C) and Data Link Nodes	Document (U) Title (U) Abstract (U)
4. AUTHOR(S)	5. CORPORATE AUTHOR
Derek Dominish, Peter Temple, Christos Sioutis, Callum Baillie and Kate Foster	DSTO Defence Science and Technology Organisation 506 Lorimer St Fishermans Bend Victoria 3207 Australia

6a. DSTO NUMBER	6b. AR NUMBER	6c. TYPE OF REPORT	7. DOCUMENT DATE
DSTO-TR-2567	AR-015-030	Technical Report	April 2012

8. FILE NUMBER	9. TASK NUMBER	10. TASK SPONSOR	11. NO. OF PAGES	12. NO. OF REFERENCES
2011/1133619/1	CIO 07/042	A/DGEA (CIO) Director General Enterprise Architecture	37	26

DSTO Publications Repository	14. RELEASE AUTHORITY
http://dspace.dsto.defence.gov.au/dspace/	Chief, Air Operations Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SA 5111

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CITATION IN OTHER DOCUMENTS Yes

18. DSTO RESEARCH LIBRARY THESAURUS

Network centric warfare, airborne mission systems, software architecture, middleware, tactical data links

19. ABSTRACT

This report discusses the Air Operations Division contribution to the Net Warrior demonstration event NW-D10. NW-D10 was held in September 2010 and demonstrated information interoperability of middleware technologies in dynamic environments with real mission systems. An overview of the NW-D10 event is provided along with a discussion of the technologies demonstrated. The outcomes of NW-D10 are presented and future Net Warrior events and enabling research are outlined.